

FIG. 1A

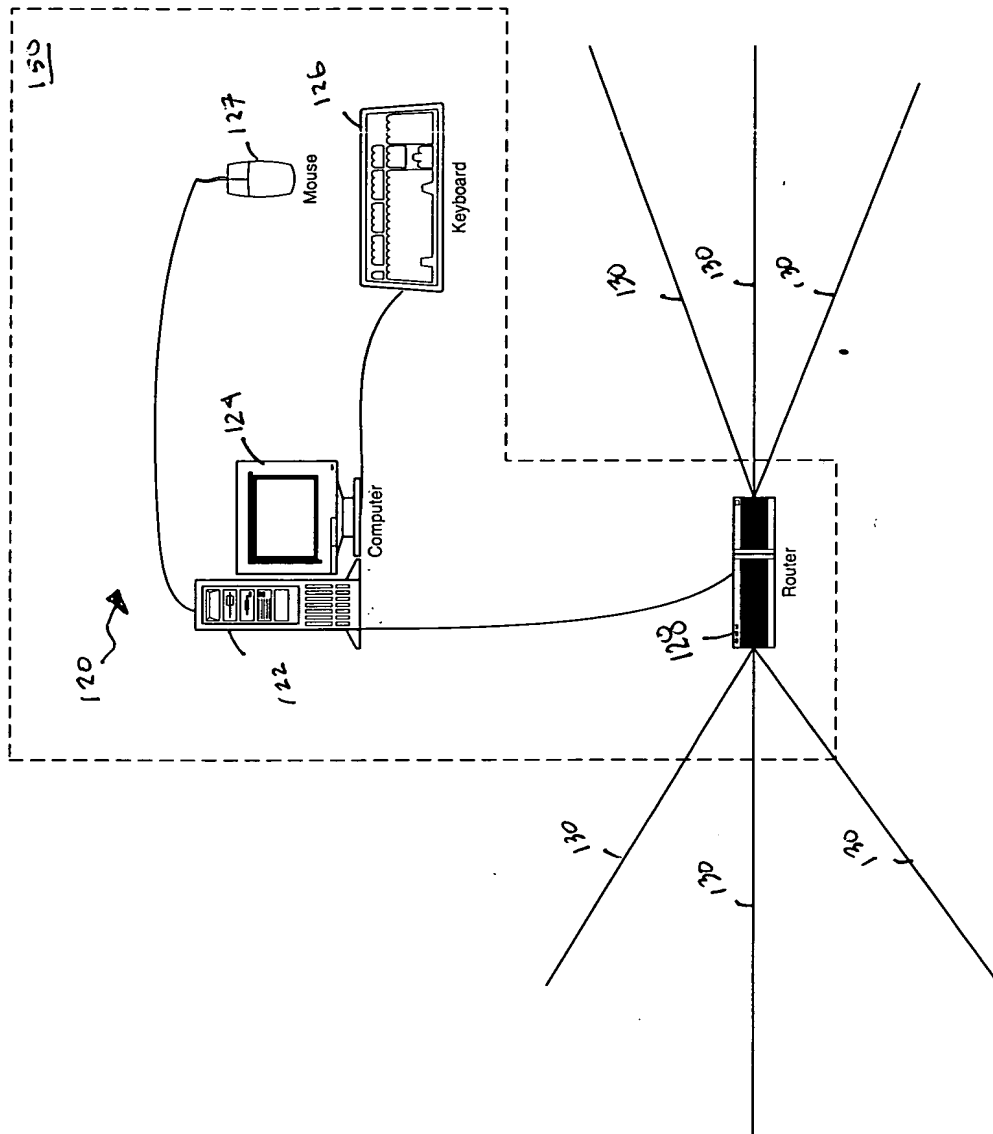


FIG. 10

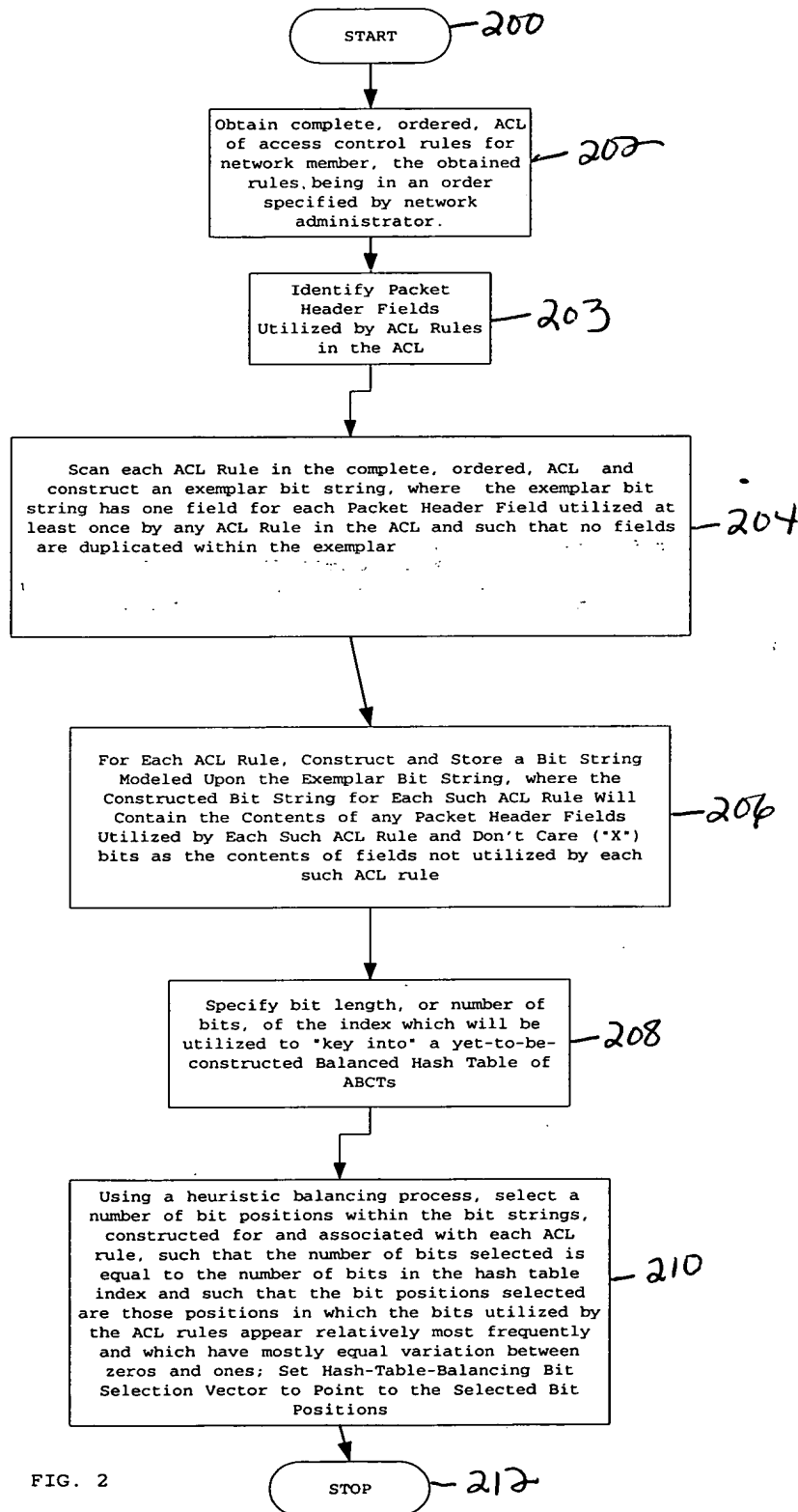


FIG. 2

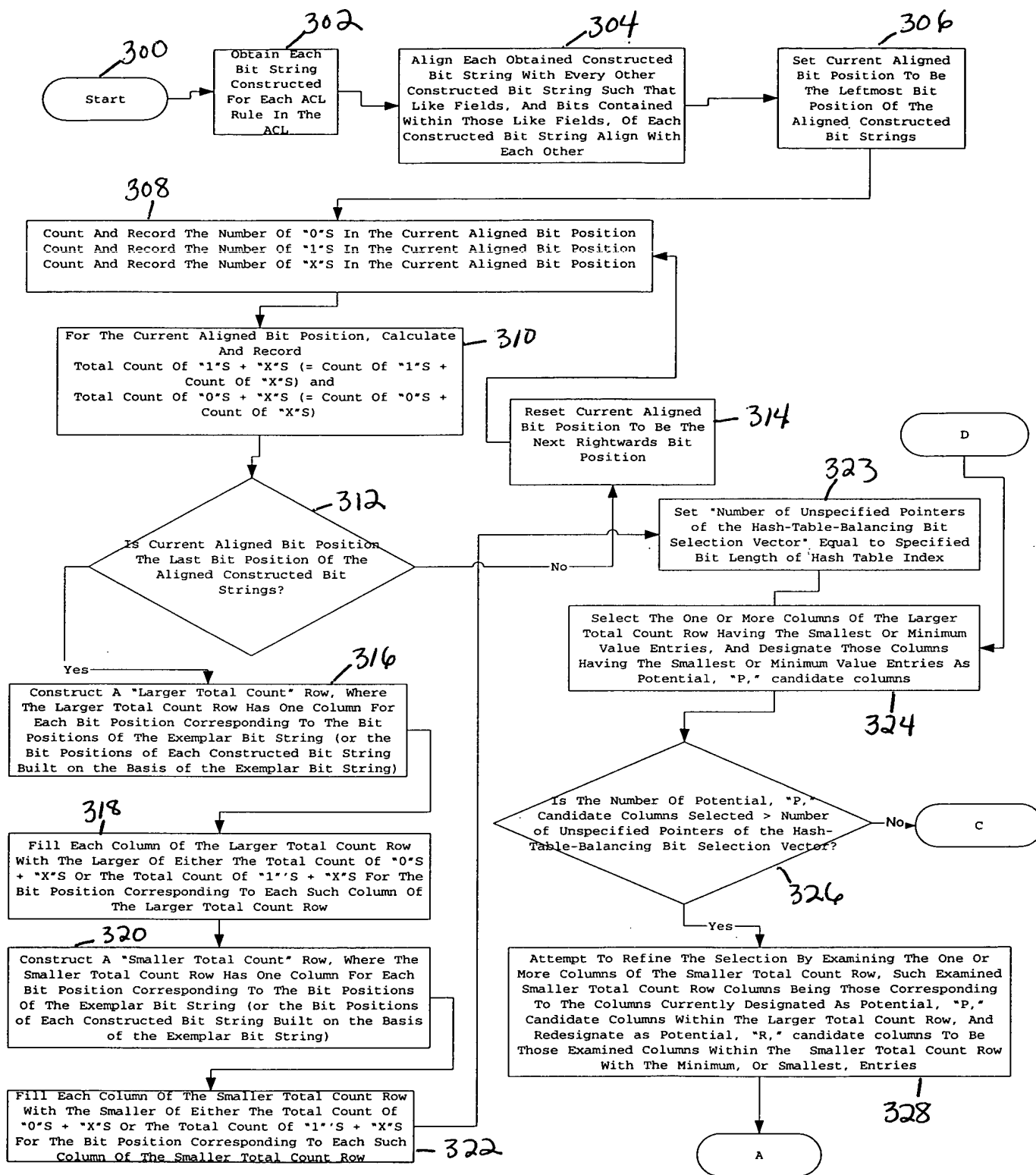
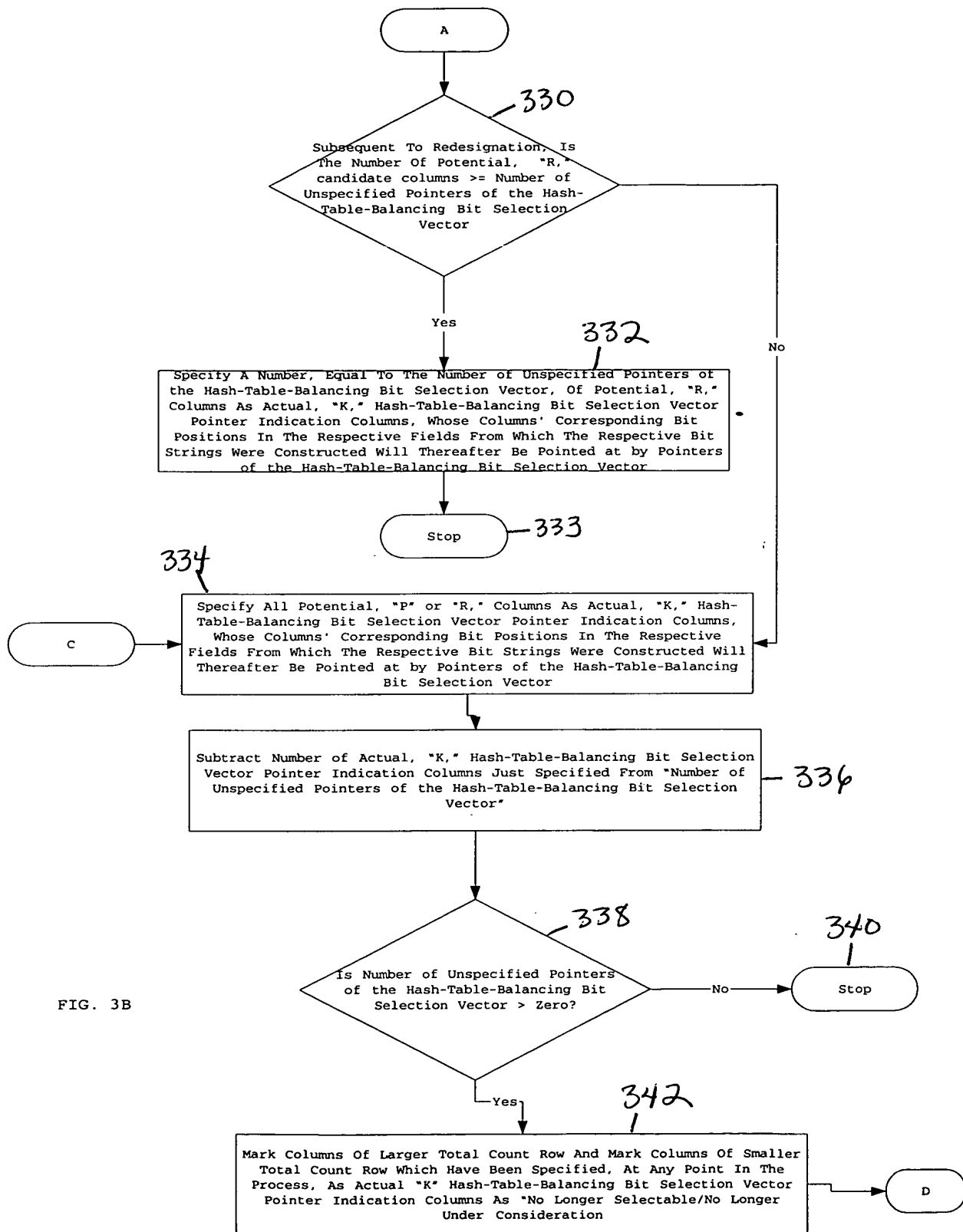
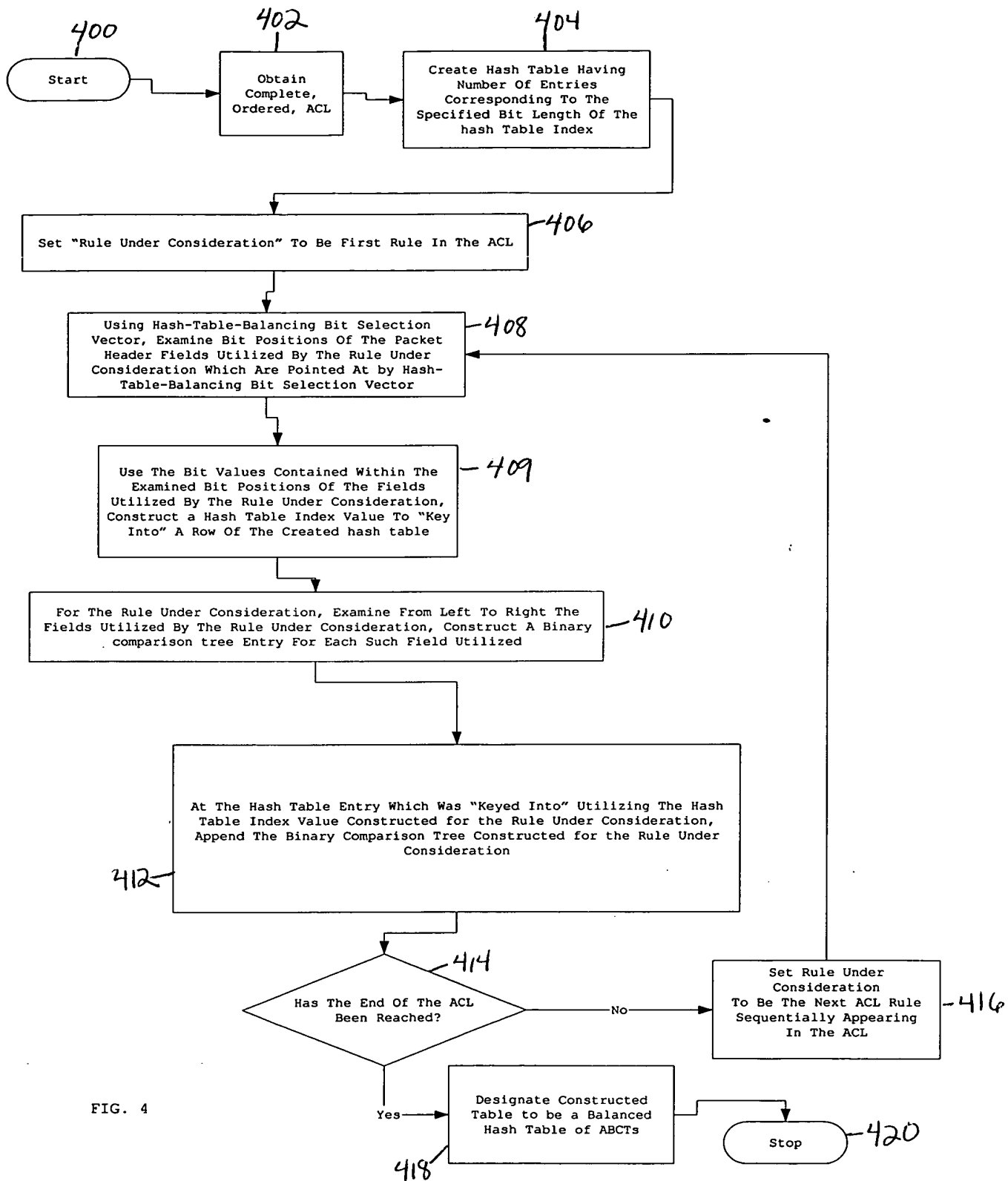
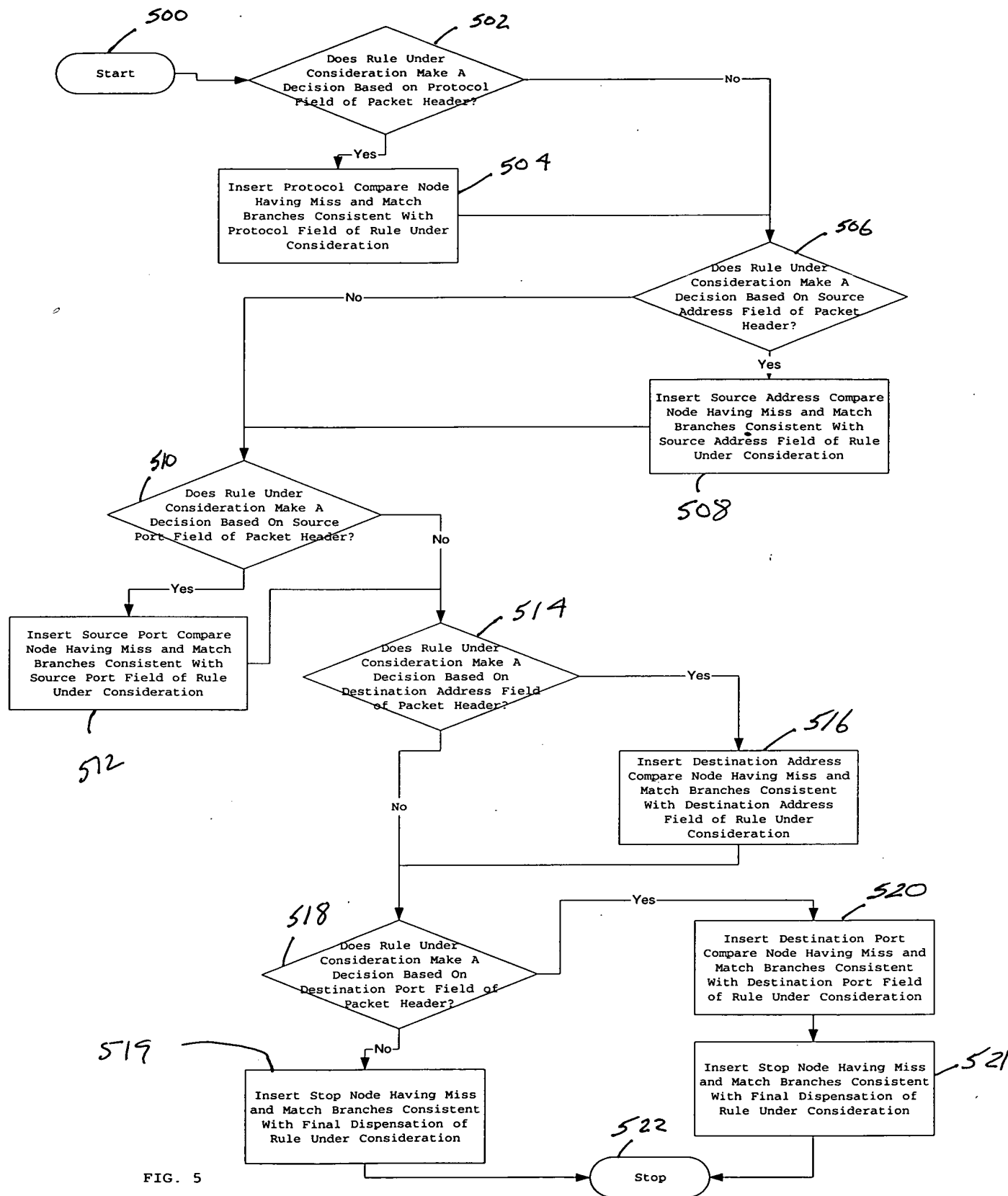


FIG. 3A







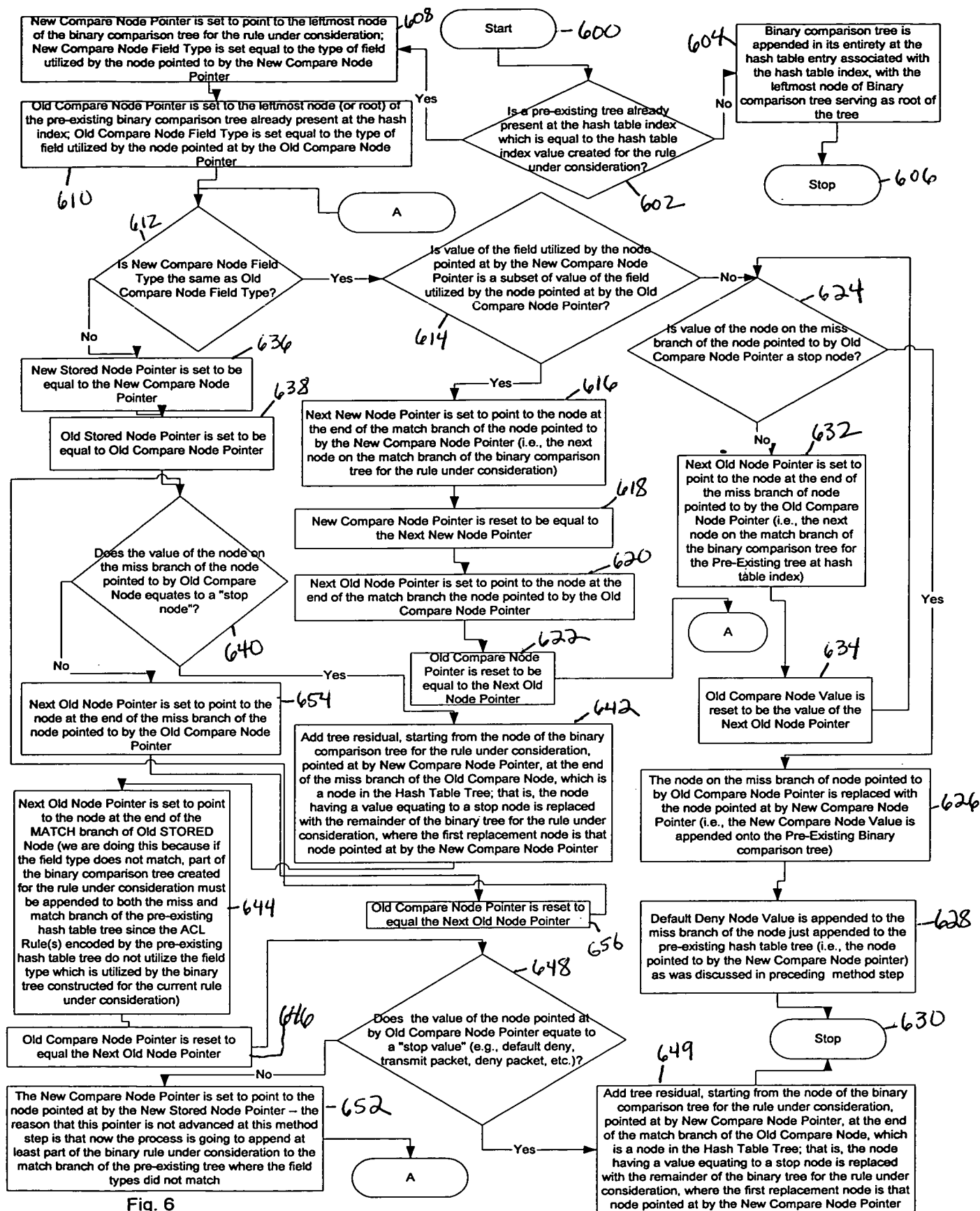


Fig. 6

Simplified Example of Ordered ACL Rule Set Typically Entered by a Network Administrator

ACL Rules in an Ordered ACL Rule Set Expressed as Plain English Statements	Examples of Coded Versions of ACL Rules Which Are Typically Utilized Within an ACL Rule Set
Permit TCP protocol packets from any source IP address going to host having an IP address of 28.16.31.10 and a port identifier equal to 28.	PERMIT TCP ANY HOST 28.16.31.10 EQ 28
Deny TCP protocol packets from any source IP address going to host having an IP address of 28.16.31.10 and a port identifier greater than 23.	DENY TCP ANY HOST 28.16.31.10 GT 23
Deny UDP protocol packets from any source IP address going to host having an IP address of 30.22.12.5 and a port identifier equal to 11.	DENY UDP ANY HOST 30.22.21.5 EQ 11
Permit UDP protocol packets from any source IP address going to host having an IP address of 30.22.12.X, where X indicates any number, or "don't care".	PERMIT UDP ANY HOST 30.22.21.X
Deny all packets having source IP address of 23.20.7.0 and any destination address (indicated by address X.X.X.X, where X indicates any number, or "don't care").	DENY TCP 23.20.7.0 X.X.X.X.
Permit TCP protocol packets from any source IP address going to host having an IP address of 28.16.31.10.	PERMIT TCP ANY HOST 28.16.31.10

Example of the Creation of an Exemplar Bit String Having One Field for Each Packet Header Field Utilized By at Least One ACL Rule in the ACL Rule Set, and the Subsequent Creation of Bit Strings for Each ACL Rule in the ACL Rule Set Based on the Created Exemplar

577321 v3

FIG. 7B

Example of the Creation of a Bit Selection Vector

"0"Count in Each Bit Position:	40440.01000.01010.11000.11111.00035.05335.02020.41313.11111
"X"Count in Each Bit Position:	00000.55555.55555.55555.55555.11111.11111.22222.33333
Total of "0" + "X" Counts:	40440.56555.56565.66555.66666.11146.16446.13131.63535.44444
"1"Count in Each Bit Position:	26226.10111.10101.00111.00000.55520.50220.53535.03131.22222
"X"Count in Each Bit Position:	00000.55555.55555.55555.55555.11111.11111.22222.33333
Total of "1" + "X" Counts:	26226.65666.65656.55666.55555.66631.61331.64645.25353.55555
Construct a "Larger Total Count" row having one row entry corresponding to each bit position in the strings which were constructed from the ACL rules; fill each row entry with the larger of either the "Total of '0' + 'X' Counts" or "Total of '1' + 'X' Counts" for the bit position corresponding to that row entry.	46446.66666.66666.66666.66666.66646.66446.64645.65555.55555
Construct a "Smaller Total Count" row having one row entry corresponding to each bit position in the strings which were constructed from the ACL rules; fill each row entry with the smaller of either the "Total of '0' + 'X' Counts" or "Total of '1' + 'X' Counts" for the bit position corresponding to that row entry.	20226.55555.55555.55555.55555.11131.11331.13131.23333.44444
Set Number of Unspecified Pointers of Bit Selection Vector = Specified Bit Length of Hash table index	Number of Unspecified Pointers of Bit Selection Vector = 4 For sake of example, assume hash table index having a bit length of 4 is specified.
Select the row entries in the "Larger Total Count" row columns having the smallest number entries; designate the bit positions corresponding to the selected row columns as potential, "P," candidate columns which might be utilized as the pointers of the Bit Selection Vector	P PP P PP P P P Note: The row columns 1, 3, 34, 39, 41, and 46 of the "Larger Total Count" row had the smallest entries (i.e., the base 10 number "4"), and thus the bit positions associated with row columns 1, 3, 34, 39, 41, and 46 of the "Larger Total Count" row are designated as potential candidate bits "P."
Since there are more Potential, "P," candidate columns than number of Unspecified Pointers of Bit Selection Vector, refine the selection by Examining the columns of the Smaller Total	R RR Note: The row columns 1, 3, and 4 of the "Smaller Total Count" row, corresponding with the selected row columns of the "Larger Total Count" row, had the smallest entries (i.e., the base 10 number "2"), and thus the bit positions associated with row columns 1, 3, and 4 of the "Smaller Total Count" row are redesignated as

Count Row, with such examined Smaller Total Count row columns being those corresponding to the larger Total Count Row columns designated as potential, "P," candidate columns; redesignate as potential, "R," candidate columns which might be utilized as the pointers of the Bit Selection Vector, those examined Smaller Total Count row columns with the smallest number entries	candidate bits "R."
Since the number or redesignated potential candidates, "R," is less than the Number of Unspecified Pointers of Bit Selection Vector, Designate all redesignated, "R," candidates as Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector	K K Note: The Number of Unspecified Pointers of Bit Selection Vector is currently equal to 4, and the number of redesignated potential candidates, "R," is 3, which is less than the Number of Unspecified Pointers of Bit Selection Vector; thus, all "R" potential candidates are specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector.
Subtract the number of Specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector, from Number of Unspecified Pointers of Bit Selection Vector	Number of Unspecified Pointers of Bit Selection Vector = Number of Unspecified Pointers of Bit Selection Vector (i.e., 4) - Number of Specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector Specified in Preceding Step (i.e., 3) = 1 pointer left unspecified
Since the number of Unspecified Pointers of Bit Selection Vector is still non-zero, Mark Specified "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by	* * Note: Row columns 1, 3, and 4 are marked with asterisks to indicate that since the these row columns have already been designated as candidates "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector.

<p>pointers of the Bit Selection Vector with asterisks indicating that such columns are no longer selectable or under consideration, since the bit positions associated with the "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector have already been specified.</p>	
<p>Thereafter, repeat the "select the row entries in the "Larger Total Count" row having smallest number entries . . ." operation above upon the row columns which have not yet been designated as candidate "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector</p>	<p>* **</p> <p>P PP P P</p> <p>Note: Row columns 1, 3, and 4 are marked with asterisks to indicate that since the bit positions associated with these row columns have already been designated as candidates.</p>

DATA OF FIG. 7C4

<p>Since there are more candidates, "P," than Number of Unspecified Pointers of Bit Selection Vector(at this point, 3 pointers have been specified as "K," meaning that one additional pointer is necessary to have the pointers required to completely point out the 4 bit hash table index), repeat the refine the selection . . . " operation above</p>	<p>R RR R R</p> <p>Note: Since all entries in the "Smaller Total Count" row columns, corresponding with the selected row columns of the "Larger Total Count" row, were the same number (i.e., the base ten number "3"), all P row columns are redesignated as candidates "R."</p>
<p>Since after redesignation there are still more candidates "R" than the Number of Unspecified Pointers of Bit Selection Vector, all "R" candidates are deemed equally good choices; consequently, the number of Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector necessary to completely point out the hash table index value (i.e., in the present example, one more pointer is needed) may be selected at random from the designated "R" row columns.</p>	<p>K</p> <p>Note: Select row column 34 at random.</p>
<p>There are now specified Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector necessary to completely point out the hash table index value (i.e., in the present example, one more pointer is needed) may be selected at random from the designated "R" row columns.</p>	<p>K KK K</p> <p>Note: These Actual, "K," Bit Selection Vector Pointer Indication Columns, whose corresponding bit positions in the respective fields from which the respective bit strings were constructed will thereafter be pointed at by pointers of the Bit Selection Vector indicate that the first, third, and fourth leftmost bit positions within the "protocol ID" field, and the fourth leftmost bit positions within the "destination address" field will be utilized as the hash table index bits.</p>

the hash table, have been fully specified.	
Definition of the Bit Selection Vector	<p>Bit Selection Vector =</p> <p>[pointer to first leftmost bit position within the "protocol ID" field;</p> <p>pointer to third leftmost bit position within the "protocol ID" field;</p> <p>pointer to fourth leftmost bit position within the "protocol ID" field;</p> <p>pointer to fourth leftmost bit position within the "destination address" field]</p>

Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees

Example Showing the Creation of a Binary Comparison Tree for First In Sequence ACL Rule in Rule Set

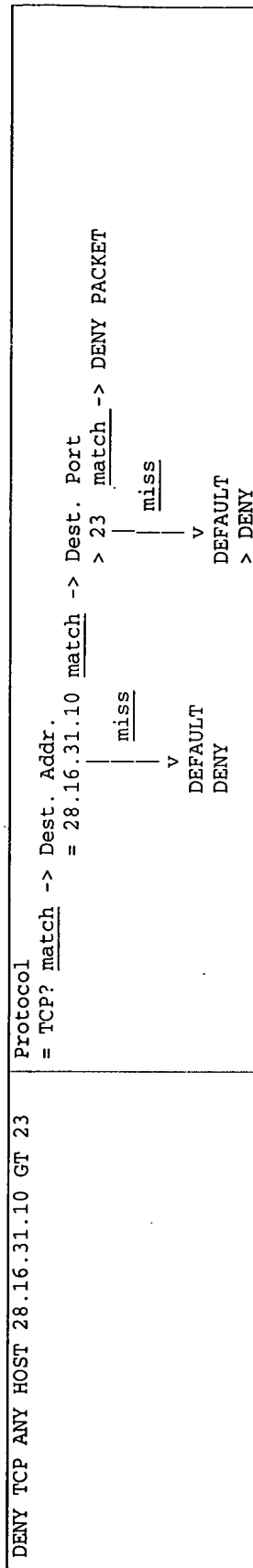
PERMIT TCP ANY HOST 28.16.31.10 EQ 28	<pre> Protocol = TCP? <u>match</u> -> Dest. Addr. = 28.16.31.10 <u>match</u> -> Dest. Port = 28 <u>match</u> -> PERMIT PACKET <u>miss</u> v DEFAULT DENY <u>miss</u> v DEFAULT DENY </pre>
---------------------------------------	--

Example Showing the Addition of a Binary Comparison Tree Constructed for the First in Sequence Rule In ACL Rule Set Into The Hash Table

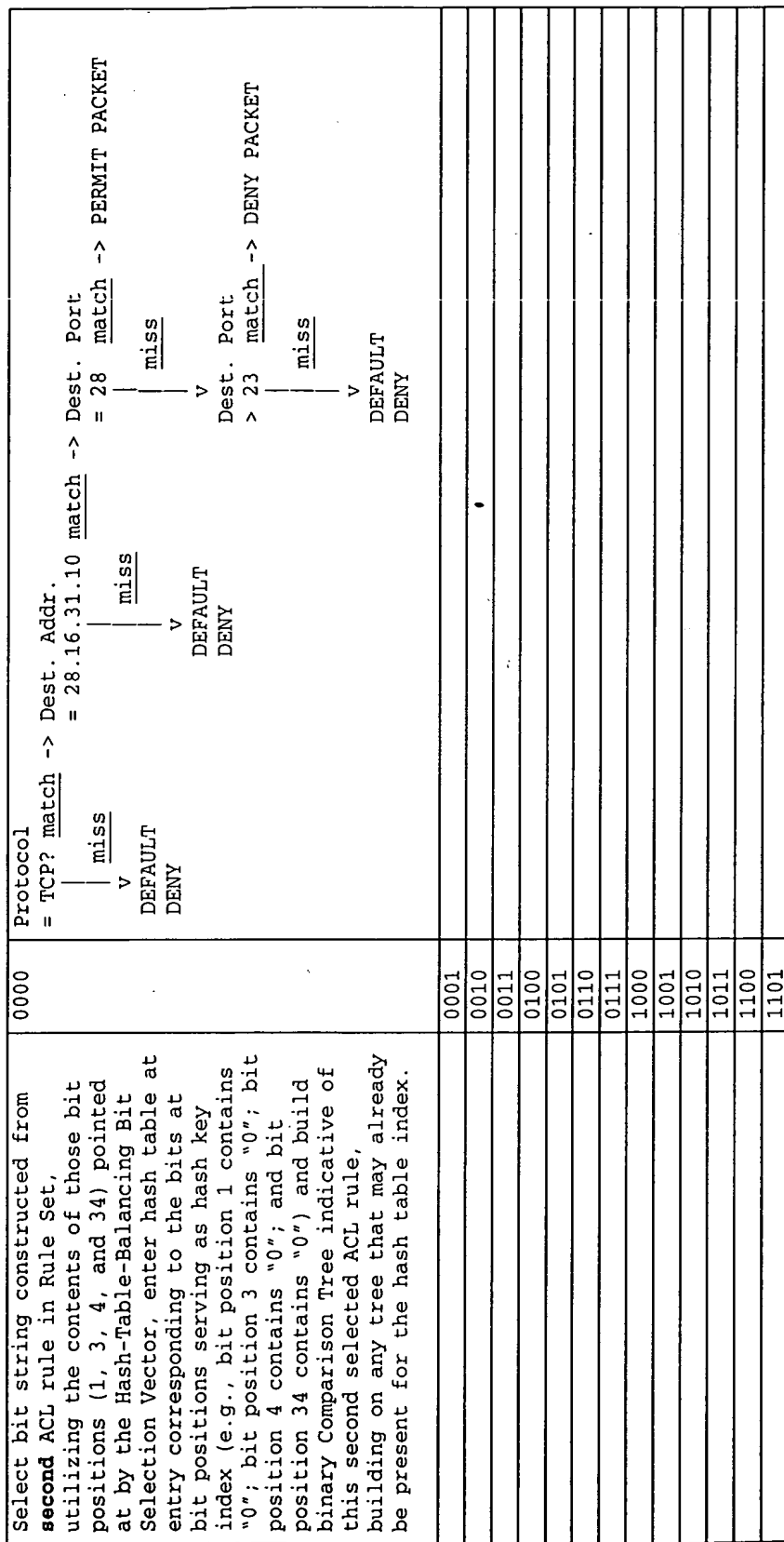
<p>Select bit string constructed from first ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "0"; bit position 3 contains "0"; and bit position 4 contains "0"; and build Binary Comparison Tree indicative of this first selected ACL rule</p>	0000	<p>Protocol = TCP? <u>match</u> -> Dest. Addr. = 28.16.31.10 <u>match</u> -> Dest. Port = 28 <u>match</u> -> PERMIT PACKET</p>
	0001	
	0010	
	0011	
	0100	
	0101	
	0110	
	0111	
	1000	
	1001	
	1010	
	1011	
	1100	
	1101	
	1110	

Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)

Example showing the creation of a Binary Comparison Tree for Second In Sequence Rule in Rule Set



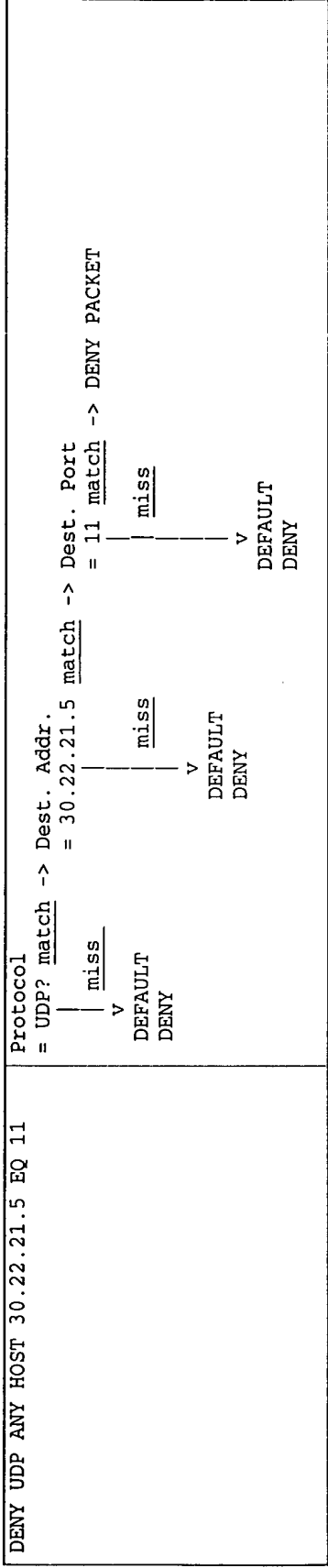
Example Showing the Addition of a Binary Comparison Tree Constructed for the Second In Sequence Rule In ACL Rule set Into The Hash Table



DATA TABLE

	1110	
	1111	

Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)
Example Showing the Creation of a Binary Comparison Tree for Third In Sequence ACL Rule in Rule Set

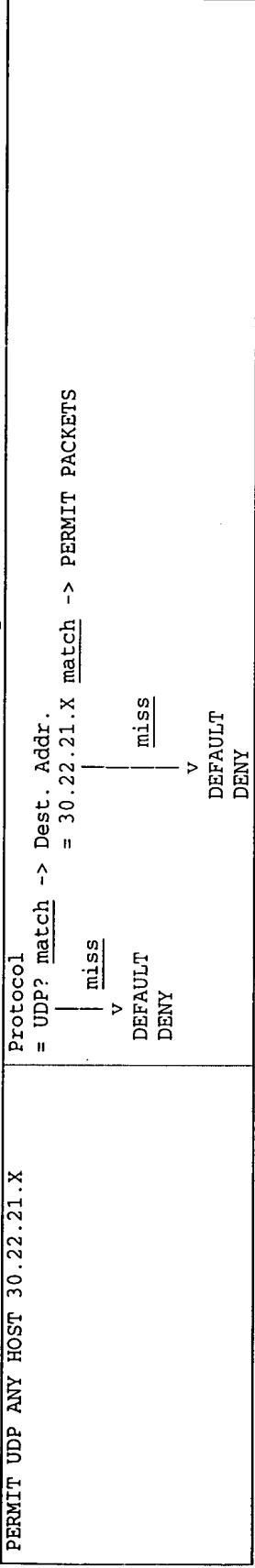


Example Showing the Addition of a Binary Comparison Tree Constructed for the Third In Sequence Rule In ACL Rule set Into The Hash Table

	0000	<p>Protocol = TCP? <u>match</u> -> Dest. Addr. = 28.16.31.10 <u>match</u> -> Dest. Port = 28 <u>match</u> -> PERMIT PACKET</p> <pre>graph TD A[Protocol = TCP? match -> Dest. Addr. = 28.16.31.10 match -> Dest. Port = 28 match -> PERMIT PACKET] -- miss --> B[DEFAULT DENY] A -- v --> C[miss] C -- v --> D[Dest. Port > 23 match -> DENY PACKET] C -- miss --> E[DEFAULT DENY]</pre>
	0001	
	0010	
	0011	
	0100	
	0101	
	0110	
	0111	
	1000	
	1001	
	1010	
	1011	

Example Showing the Construction of Balanced Hash Table Of ACL Binary Comparison Trees (cont.)

Example Showing the Creation of a Binary Comparison Tree for Fourth In Sequence ACL Rule in Rule Set



Example Showing the Addition of a Binary Comparison Tree Constructed for the Fourth in Sequence Rule In ACL Rule set Into The Hash Table

	0000	Protocol = TCP? <u>match</u> -> Dest. Addr. = 28.16.31.10 <u>match</u> -> Dest. Port = 28 <u>match</u> -> PERMIT PACKET v DEFAULT DENY
	0001	
	0010	
	0011	
	0100	
	0101	
	0110	
	0111	
	1000	
	1001	
	1010	
	1011	
	1100	
	1101	
	1110	

<p>Select bit string constructed from fourth ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "1"; bit position 3 contains "1"; bit position 4 contains "1"; and bit position 34 contains "1") and build binary Comparison Tree indicative of this fourth selected ACL rule, building on any tree that may already be present for the hash table index</p>	1111	<p>Protocol = UDP? <u>match</u> -> Dest. Addr. <u>miss</u> v DEFAULT DENY</p> <p>= 30.22.21.5 <u>match</u> -> Dest. Port <u>miss</u> v DEFAULT DENY</p> <p>= 11 <u>match</u> -> DENY PACKET</p> <p>= 30.22.21.X <u>match</u> -> PERMIT PACKET <u>miss</u> v DEFAULT DENY</p>
--	------	--

Example Showing the Construction of Balanced Hash Table Of ACL Binary Comparison Trees (cont.)

Example Showing the Creation of a Binary Comparison Tree for Fifth In Sequence ACL Rule in Rule Set

DENY TCP 23.20.7.0 X.X.X.X.	Protocol = TCP? <u>match</u> -> Source Addr. <u>miss</u> <u>miss</u> v v v DEFAULT DEFAULT DENY DENY
-----------------------------	--

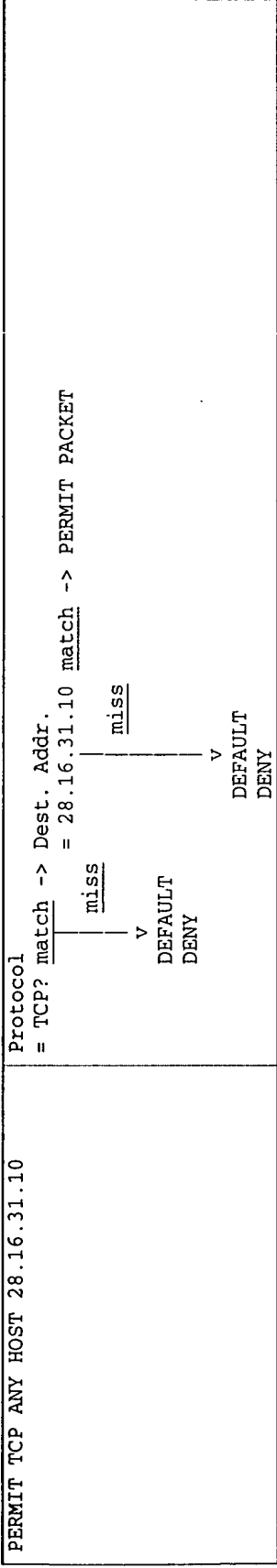
Example Showing the Addition of a Binary Comparison Tree Constructed for the Fifth In Sequence Rule In ACL Rule set Into The Hash Table

<p>Select bit string constructed from fifth ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "0"; bit position 3 contains "0"; bit position 4 contains "0"; and bit position 34 contains "X") and build binary Comparison Tree indicative of this fifth selected ACL rule, building on any tree that may already be present for the hash table index; however, since bit at bit position 34 is X, the rule will be appended at both 0000 and 0001, since bit position 34 may be either 0 or 1. In addition, since the rule itself applies to any destination address, the miss branch of all destination branches present must feed back into the source address compare instruction associated with this Fifth Rule.</p>	0000	Protocol = TCP? <u>match</u> -> Dest. Addr. <u>miss</u> <u>miss</u> <u>miss</u> v v v v DEFAULT DEFAULT DEFAULT DENY DENY DENY DENY Dest. Port > 23 <u>match</u> -> DENY PACKET <u>miss</u> v -> Srce. Addr. = 23.20.7.0 <u>match</u> -> DENY PACKET <u>miss</u> v DEFAULT DENY -> Srce. Addr. = 23.20.7.0 <u>match</u> -> DENY PACKETS <u>miss</u> v DEFAULT DENY
--	------	--

<p>Select bit string constructed from fifth ACL rule in Rule Set, utilizing the contents of those bit positions (1, 3, 4, and 34) pointed at by the Hash-Table-Balancing Bit Selection Vector, enter hash table at entry corresponding to the bits at bit positions serving as hash key index (e.g., bit position 1 contains "0"; bit position 3 contains "0"; bit position 4 contains "0"; and bit position 34 contains "X") and build binary Comparison Tree indicative of building on any tree that may already be present for the hash table index; however, since bit at bit position 34 is X, the rule will be appended at both 0000 and 0001, since bit position 34 may be either 0 or 1.</p>	<p>0001</p> <p>Protocol = TCP? <u>match</u> -> Source Addr. <u>miss</u> v <u>miss</u> DENY <u>miss</u> v DEFAULT DENY</p>
	0010
	0011
	0100
	0101
	0110
	0111
	1000
	1001
	1010
	1011
	1100
	1101
	1110
	1111
	<p>Protocol = UDP? <u>match</u> -> Dest. Addr. <u>miss</u> v <u>miss</u> DEFAULT <u>miss</u> DENY <u>miss</u> v = 30.22.21.X <u>match</u> -> PERMIT PACKET <u>miss</u> v = 30.22.21.5 <u>match</u> -> Dest. Port = 11 <u>match</u> -> DENY PACKET</p>

Example Showing the Construction of Balanced Hash Table of ACL Binary Comparison Trees (cont.)

Example Showing the Creation of a Binary Comparison Tree for Sixth In Sequence ACL Rule in Rule Set



Example Showing the Addition of a Binary Comparison Tree Constructed for the Sixth In Sequence Rule In ACL Rule set Into The Hash Table

